

FYP

**Simulation of Automated Personalized Payment and Notification Application
(SAPPNA)**

by

Nurfhara Nadia Binti Nordin
16381

Dissertation submitted in partial fulfillment of
the requirements for the
Bachelor of Technology (Hons)
(Business Information System)

JANUARY 2015

Universiti Teknologi PETRONAS
32610 Bandar Seri Iskandar
Perak Darul Ridzuan

CERTIFICATION OF APPROVAL

FYP

**Simulation of Automated Personalized Payment and Notification Application
(SAPPNA)**

by

Nurfhara Nadia Binti Nordin
16381

Dissertation submitted in partial fulfilment of
The requirements for the
Bachelor of Technology (Hons)
(Business Information System)

Approved by,

(Associate Professor Dr. Dhanapal Durai Dominic)

CERTIFICATION OF ORIGINALITY

This is to certify that I am responsible for the work submitted in this project, that the original work is my own. The original work contained herein have not been undertaken or done by unspecified sources or persons.

NURFHARA NADIA BINTI NORDIN

ABSTRACT

The main objective of this project is to develop a mobile payment that can be used to ease the user life as there been a studied about how important to know the efficiency of the mobile payment application in terms of reliability and usability. Besides, to implement and develop an android mobile application that can be used by the user. The problem happens when people busy with the work life, they tend to forget to pay their bills. For example, electricity bill was not paid then the electricity board will cut off and affect the daily life process.

From the findings, it will be develop on mobile application and using the Android as operating system. As most of the user has their own smartphone and it much more intimated devices compare to the laptops or desktop. For the Android operating system, it is most popular mobile operating system platform among the smartphone user.

The scopes of study for Simulation of Automated Personalized Payment and Notification Application (SAPPNA) are first, automatic transaction payment that simulate the automatic transaction of money to the vendor without require notification from the user. Second, notification to the user to notify the bills already paid and inform if the amount of money not sufficient to pay bills.

The methodology used is prototype model and the idea of choosing prototyping model because as the project still on going, it ensures mores changes can be made in order to meet the user requirement for the final product. For future enhancement, this application will have system administrator that will supervise and handle the security as well the database and also payment gateway that will do actual transaction payment with bank. In conclusion, the application has proved that it can help and solve the problem that had been faced by the user.

ACKNOWLEDGEMENT

First and foremost, thanks to Almighty Allah for giving me the strength and spirit to do and work with this project. He is the One who made everything happen according to plan even though there a problem.

I would like to express my gratitude for getting an experienced supervisor, Associate Professor Dr. Dhanapal Durai Dominic. I am also grateful to have him as my supervisor who always been there when I need advice and opinion. The supervision that he gave truly helped the progression and smoothness of my final year project.

Not to forget, thank you to all lecture for their help by giving the guidance in the areas need to be cove in the final year project. Last but not least, special thanks to my family and friends who always been there through hard and down. All the guidance, pray and support are highly appreciated.

TABLE OF CONTENTS

CERTIFICATION	i
ABSTRACT	iii
ACKNOWLEDGEMENT	iv
CHAPTER 1: INTRODUCTION	1
1.1 Background Study	1
1.2 Problem Statement	1
1.2.1 Problem Identification	1
1.2.2 Significant of the Project	2
1.3 Objectives of the Study	2
1.4 Scope of Study	3
1.5 Feasibility Study	3
1.5.1 Technical Feasibility	3
1.5.2 Economy Feasibility	3
1.5.3 Operational Feasibility	3
CHAPTER 2: LITERATURE REVIEW	4
2.1 Introduction to Mobile Payment	4
2.2 Traditional of Payment Method	5
2.3 Related Works	5
2.4 What is Mobile Application?	7
2.5 Why Create Android Application?	7
CHAPTER 3: METHODOLOGY	9
3.1 Data Collection Model	9
3.1.1 Literature Review	9
3.1.2 Questionnaire	9
3.1.3 Existing Mobile Applications	14
3.2 System Development	15
3.2.1 Analysis Phase	16
3.2.2 Design and Development Phase	16
3.2.3 Implementation Phase	17
3.2.4 Prototype Phase	19
3.3 System Architecture	19
3.4 System Flow	20
3.4.1 Use Case Diagram	20
3.4.2 Entity Relationship Diagram	22
3.4.3 Data Flow Diagram	23
CHAPTER 4: RESULTS AND DISCUSSION	25
4.1 System Usability Scale Test	25
4.2 Interface Design	31
CHAPTER 5: CONCLUSION	34
REFERENCES	35
APPENDICES	37

LIST OF FIGURES

FIGURE 1. Result for the age group	10
FIGURE 2. Results for the gender	10
FIGURE 3. Result for Occupational Status	11
FIGURE 4. Result for smartphone user	11
FIGURE 5. Result for type of smartphone	12
FIGURE 6. Results for Operating System used by the user	12
FIGURE 7. Result does the user use smartphone for payment transaction	13
FIGURE 8. Result does user forget to pay their bills	13
FIGURE 9. Result finding do really user need to be remind to pay the bills	14
FIGURE 10. Prototype Model	15
FIGURE 11. System Architecture for SAPPNA	19
FIGURE 12. Use Case Diagram of SAPPNA	21
FIGURE 13. Entity Relationship Diagram of SAPPNA	22
FIGURE 14. Data Flow Diagram of SAPPNA	23
FIGURE 15. Results for the visibility of system status	26
FIGURE 16. Results for match between system and the real world	26
FIGURE 17. Results for user control and freedom	27
FIGURE 18. Results for consistency and standards	27
FIGURE 19. Results for error prevention	28
FIGURE 20. Results for recognition or recall	28
FIGURE 21. Results for flexibility and efficiency of use	29
FIGURE 22. Results for a aesthetic and minimalist design	29
FIGURE 23. Results for help users recognize diagnose and recover from errors	30
FIGURE 24. Results for help and documentation	30
FIGURE 25. The User Home	31
FIGURE 26. The Home Page	32

FIGURE 27. The Login Page	32
FIGURE 28. The Register Page	33
FIGURE 29. The Add Bill Page	33

LIST OF TABLES

TABLE 1. Table of Comparison between the Existing Applications	5
TABLE 2. Results from the Unit Testing	17

CHAPTER 1

INTRODUCTION

1.1 Background of Study

Today the world is moving towards mobile computing, all the work and things need to be done through mobile phones or intelligent devices are increasing day by day. As a result, the easiness of doing the task is also increasing. As we know, with the advance technology, the mobile phones are equipped with much functionality which surpass telephony needs, and which inspire the development of value-added mobile service, the use of mobile phones as access devices, and mobile commerce in general.

One of the things need to be done is the mobile payment. Pousttchi (2003) mentioned that mobile payment (MP) as the type of payment transaction processing in the course of which within the electronic procedure where the user use mobile communication technique in conjunction with mobile devices for initiation, authorization or realization payment.

1.2 Problem Statement

1.2.1 Problem Identification

Because of busy life in real-time, people tend to forget to pay their bills for telephone, water, electricity and others. These small things if people tend to forget, it could affect the daily life process. For example the electricity bills,

if the bill was not paid and electricity board being cut off, it will affect the daily life process because no electricity. Basically we will depend on the electricity to ensure the work to be done.

1.2.2 Significant of the Project

The SAPPNA will be able to give an alarm to the user when they need to pay their bills when the will be the dateline. Besides, SAPPNA also will help the user to pay automatically their bills by integrate to bank account. It will deduct and pay their bills automatically once the bill needs to be pay. Lastly, if the amount of money in the account is not sufficient, SAPPNA will send a notification to inform the user that the balance in the account is not sufficient enough to pay the bills.

1.3 Objectives of the Study

The objectives of the project are as follows:

1. To study how a mobile payment can be used to ease the user life

A study need to be done to understand and differentiate the style of using the mobile as a method of payment. It is important to know the efficiency of the mobile payment application in terms of reliability and usability. It is belief that the market of mobile payment has now entering into the new dimensions and being used to replace the regular payment methods (Rehman & Coughlan, 2013).

2. To develop and implement an android mobile application that can be used by the user for the application

Findings on the implementation of SAPPNA for the user has be done. Results were measure for improvement and future reference. The mobile application was develop base on the research data found in the early stage of the project as well as it include previous research and findings.

1.4 Scope of Study

The project needs to have the positive impact to it surrounding during its implementation. Hence, this project only focuses on the **automatic transaction payment** and **notification to the user**. For automatic transaction payment, it help the user to pay the bills automatically without require an approval from the user. For notification to the user, will be notify once the payment already made and will be inform if the money not sufficient to pay the bills.

1.5 Feasibility of Study

Before it begins, the feasibility studies are needed in place to guarantee that the study is conceivable and smoothen the improvement cycle of the study.

1.5.1 Technical Feasibility

In the technical point of view, the study is feasible. The technical part of the study happen in the implementation phase in the System Development Life Cycle where the study is relied upon to build up a mobile application for the automated personalize payment.

1.5.2 Economy Feasibility

The research and development of this project will be cost benefit. The cost will be properly manageable in order to have an efficient payment transaction application.

1.5.3 Operational Feasibility

In operational feasibility, it will divide into two. First, the availability of software, hardware and database. Second, the skill level of the operational which will be the user.

CHAPTER 2

LITERATURE REVIEW

2.1 Introduction of Mobile Payment

Mid-1990s has been become the point where a lot of focus to use mobile phones for the financial transaction process. This type of processing known as *mobile payments* or *m-payments*. (Pousttchi, 2007). Mobile payments or m-payments are defined as a type of payment financial transaction process using mobile devices or in other hand using the mobile communication (Pousttchi, 2007). The techniques include the initiation, authorization, or completion of payment that can add more value in the mobile payment service.

The mobile payment service is a process aim to reduce the cost and time of generating public key operation and registration process in mobile payment, the system should have three stakeholders which include the consumer (client), the merchant and the issuer. The client exchanges the message with the issuer through the merchant to satisfy the integrity and other aspects of mobile transaction and complete the transaction. Once the client has bought product or received the service, the client will send the message to the issuer through the merchant and then the client will get the confirmation message of the successful transaction.


2.2 Traditional of Payment Method

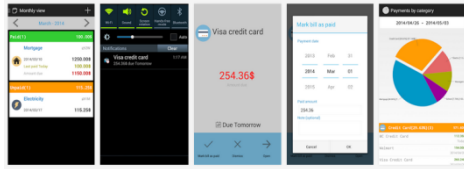
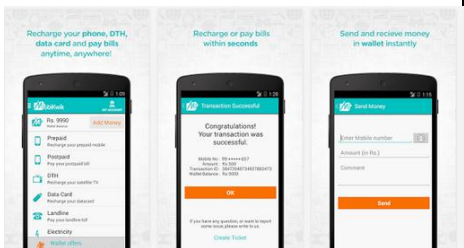
The history of purchasing finances originates with the simple barter in which two parties' exchanges any goods, resources or service for any mutual benefits. In this section is to understand how the payment process traditionally before the existing of e-payment and any application for the payment system. Previously, the payment method was money, pay checks, and credit. In early of 1200BC, shells were used as form of money until 1000 BC (Brown, 2006). Pay checks or known as pay cheque is a form of paper where it issued by a payer to pay to certain people and credit are cards that are very successful use for financial transaction as it renown around the globe (Brown, 2006).

2.3 Related Works

In Play Store, there are few mobile applications that are specifically created to help user in the payment transaction process.

TABLE 1. Table of Comparison between the Existing Applications

No	Name of the application	Description	Comment
1.	<p>Bill Reminder</p> <p>Refer: https://play.google.com/store/apps/details?id=com.amazier.apps.billreminder</p> 	<p>This application helps to manages bills and reminds before the due date. Some of the feature are manage all bills, friendly notification with paid/payment action, ease of access, customization, backup and restore. Bill Reminder lets the user to manage and organize all bills in one place utility bills. It help</p>	<p>This application only help the user to get notification to pay bills, and no transaction will happen. The user still need to aware that when they get the notification they also need to do the transaction in order to pay the bills.</p>

		<p>automatically posts reminder of all bills with simple payment action, making bill management easy and stress free. It also can organize bills as Upcoming, Overdue, Paid and Unpaid based on the due date and manage all the bills in easy manner.</p>	
2.	<p>Easy Bills Reminder</p> <p>Refer: https://play.google.com/store/apps/details?id=com.aa3.easybillsreminder</p> 	<p>The idea behind Easy Bills Reminder is to allow user to keep an eye on their bills by sending daily payment reminder notifications on the use device. The notifications will be sent regardless the application is open or closed or even after restarting the user device.</p>	<p>Method of payment for this application is using Visa Credit Card.</p>
3.	<p>Mobile Recharge and Bill Pay</p> <p>Refer: https://play.google.com/store/apps/details?id=com.mobikwik_new</p> 	<p>This application is small in size, fast and light on user device. Using Mobikwik wallet to pay all the services by one click. It has lots of payment options either by using credit card, debit card or net banking.</p>	<p>User need to place their money in the Mobikwik wallet to pay the bills. It will difficult for the user to pay the bills because they need to top up the account in order to pay the bills.</p>

From Table 1, all the existing application equipped with the notification function to alert the user to pay the bills. But the gap between three of existing application, it does not have automatic payment transaction in order to pay the bills.

2.4 What is Mobile Application?

Mobile application is software where now popular for user to act as tool in order to perform any tasks, activities or certain type of process. Many developers involve in developing a mobile application since everyone are using a smartphone. Significant increases of mobile application that exist in current market make it become popular among the smartphone user.

Mobile application becomes important to the user as it supported by (Jahns, 2010). Some of the reasons are the ubiquity of smartphones. The smartphone will increasingly replace feature phone in the stores. Nearly everyone will be able to use smartphone application. Next, the manifoldness of possibilities of smartphone can replace the act of laptops or desktops as some of the task can be restricted to computer or desktops.

Next, the proximity to user; Smartphone are much personal and intimate devices rather than laptops or desktops. For most users, they use their phones frequently than other devices. The attractiveness happen when the service or product being able to place as near as how close the user and smartphone. Therefore, it is hope that this project may benefit the user life.

2.5 Why create Android Application?

Android can be considered as the most popular mobile operating system platform around the globe. Millions of smartphone user used Android because of its powerful and usefulness. As it is the largest installed base of any mobile

platform and it still growing fast day by day. According to Bell, the data of user using Android has increased from 43.75% to 44.62% respectively.

There are many reasons why Android is better than iOS. This is supported also by (Ion, 2014) also found that Android is better than iOS:

- Total interface customization, enable the user to customize to interface as user see fit. The icons can move around, group them however the user like or even ban any application to the Application Drawer.
- Endless sharing option where user be able to move the content through any installed application with sharing capabilities
- Visible file system. If the user wants to connect the smartphone to the computer, the use will be able to see the file. For iOS, however users are only allowed to share data within the Apple devices only.
- Android offers many choices of widgets and applications. There are thousands of widgets and applications available from Google Play, giving the users tones of flexibility that iOS cannot match

CHAPTER 3

METHODOLOGY

3.1 Data Collection Method

There are three different methods for the author to gather all the data and information that will be used in this project. The methods are:

- Literature Review
- Survey/Questionnaires
- Existing mobile applications

3.1.1 Literature Review

The author has done some reading through the journals, article conference papers, magazines, newspapers and also books. The areas of studies include the introduction of mobile payment, traditional payment method, related works, what is mobile application and why create an android application. From the studies it help the author to know more in details how a mobile payment application able to help user to ease their life.

3.1.2 Questionnaires

The author has conducted questionnaires to obtain feedback and information regards to the application and also to identify either the user really faced the problem or not. The survey questionnaires consist of set of question where respondents can choose multiple answers. The author also used the answer from respondents to expand the current situation relate to this project.

35 numbers of respondents were recorded answered the questionnaires. The question was divided into two parts. The first part of the questions about the profile of potential user and second part focus on questioning the user about the problem that they are facing.

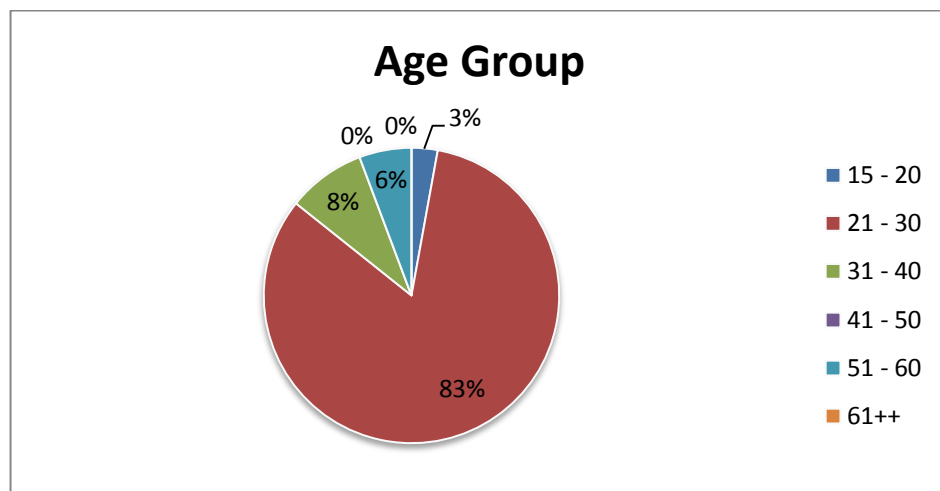


FIGURE 1. Result for the age group

The age group of respondents from Figure 1 shows that 83% age from 21 -30 years old, 8% between 31 – 40 years old, 6% between 51 – 60 years old, 3% between 15 – 20 years old. The age between 41 – 50 years old and 61 and above recorded none.

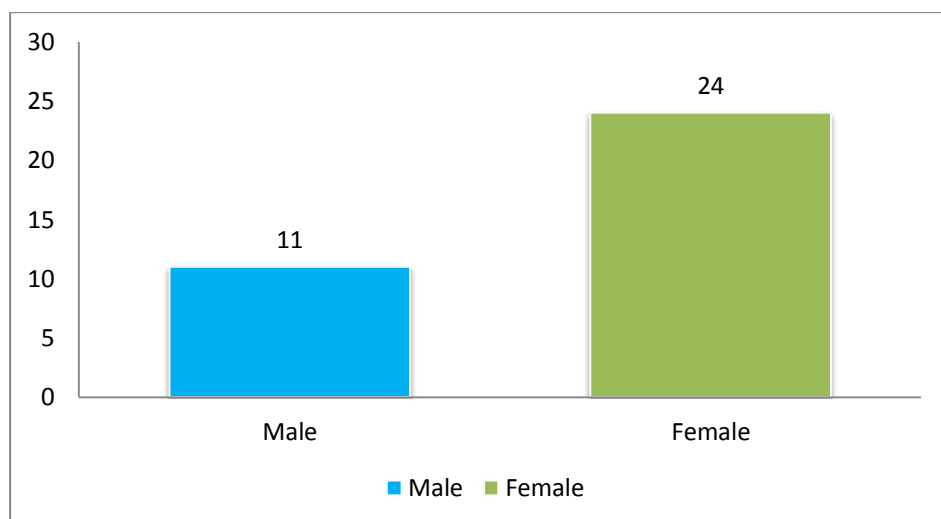


FIGURE 2. Results for the gender

For gender, the number of respondent from male only 11 and number of respondents from female are 24.

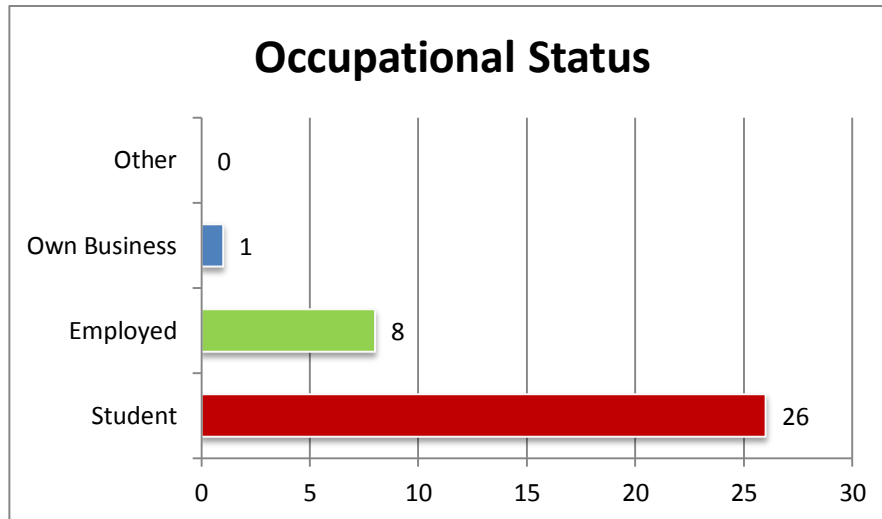


FIGURE 3. Results for Occupational Status

Respondent who answered the questionnaires basically are student, 74 % are student while 23 % are employer and 3% own a business.

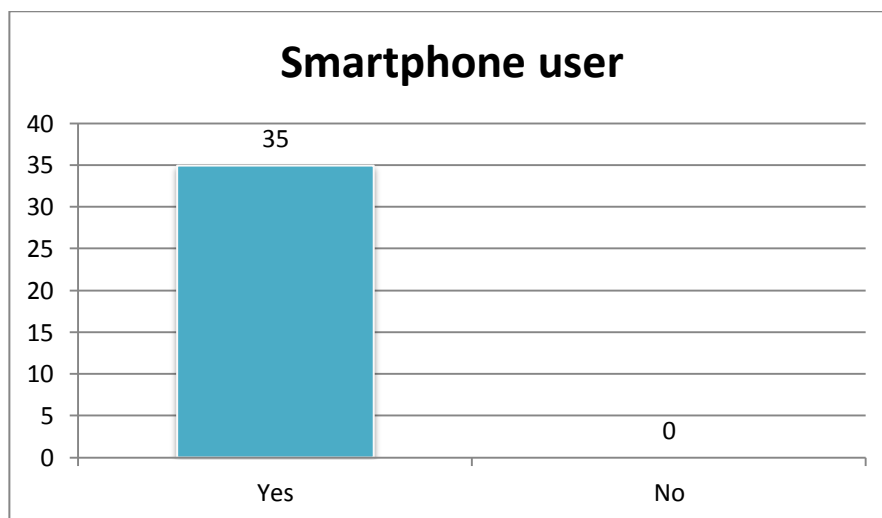


FIGURE 4. Result for smartphone user

From the survey questionnaire 100% of the respondent claims that they own smartphone and none of them do not own any smartphone at all.

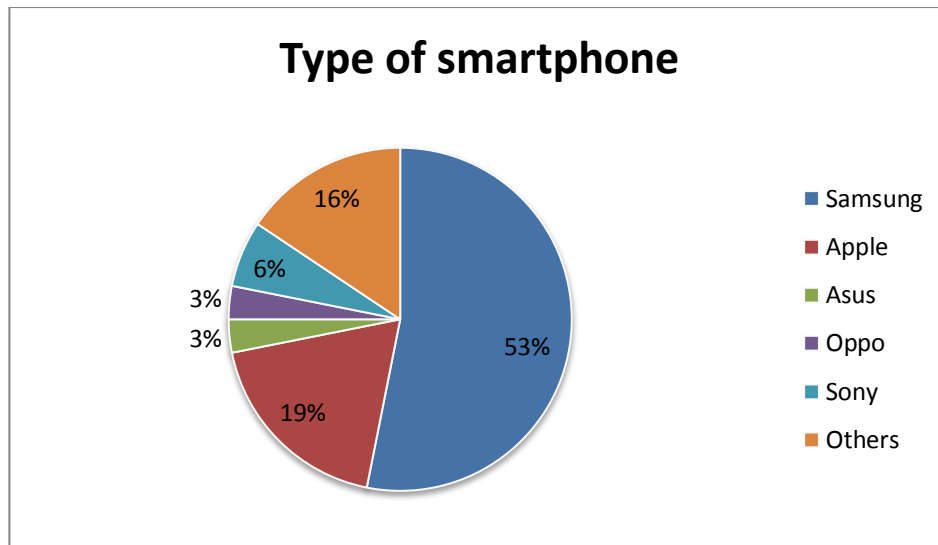


FIGURE 5. Result for type of smartphone

For type of smartphone that the respondent use, 53% were Samsung, 19% were Apple, 16% others, 6% were Sony and 3% Oppo and Asus.

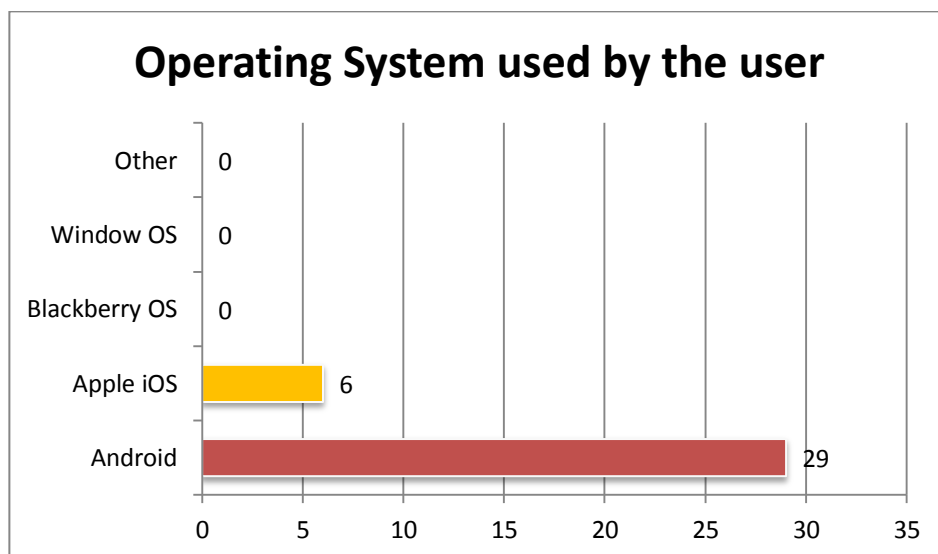


FIGURE 6. Results for Operating System used by the user

For operating system, 83% of the respondent use Android as their operating system while only 17% use Apple iOS. For other operating system none of the respondent uses it.

Based from the interview result, author beliefs that the SAPPNA has strong reason need to be built as mobile application and android as it operating system.

Besides, most of the respondents use Android as their operating system compare to the other operating system like Apple iOS, Blackberry OS, Window OS or others.

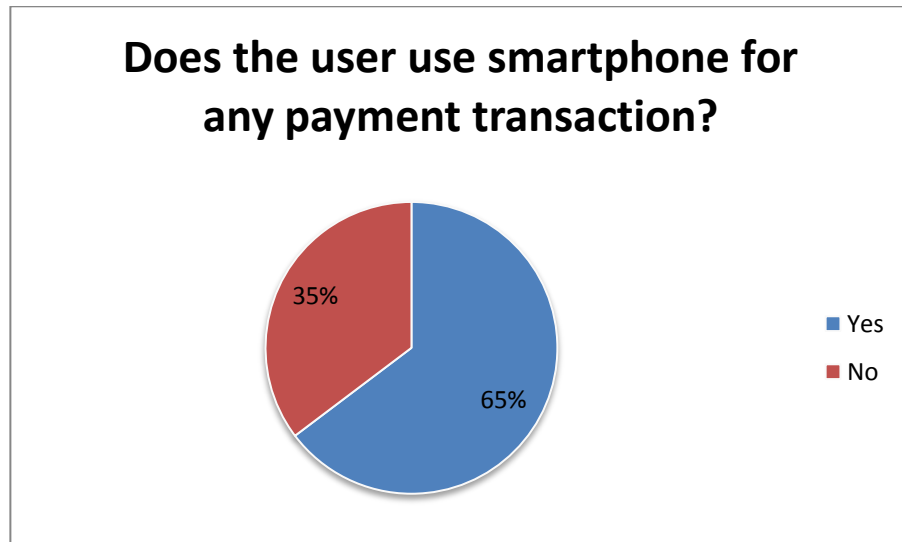


FIGURE 7. Results in understanding does the user use smartphone for payment transaction

65% out of 100% of respondents use their smartphone to do any payment transaction payment. The author belief that by using the smartphone, the process will be easier and less time taken as well energy.

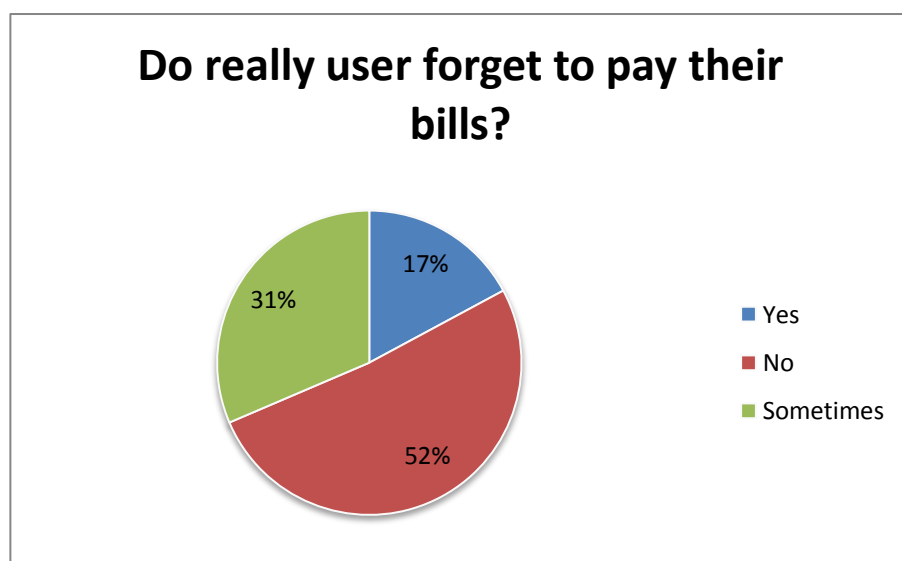


FIGURE 8. Results does user forget to pay their bills

The finding does the user really forget to pay their bills shows that 17% answering yes, 33% answering no and the remaining are sometimes. This is because most of the respondents are among the student and it can relate to the 52% answering sometimes. Other than that, the student especially does not have any big responsibilities to much kind of bills as those who are already working or has own business.

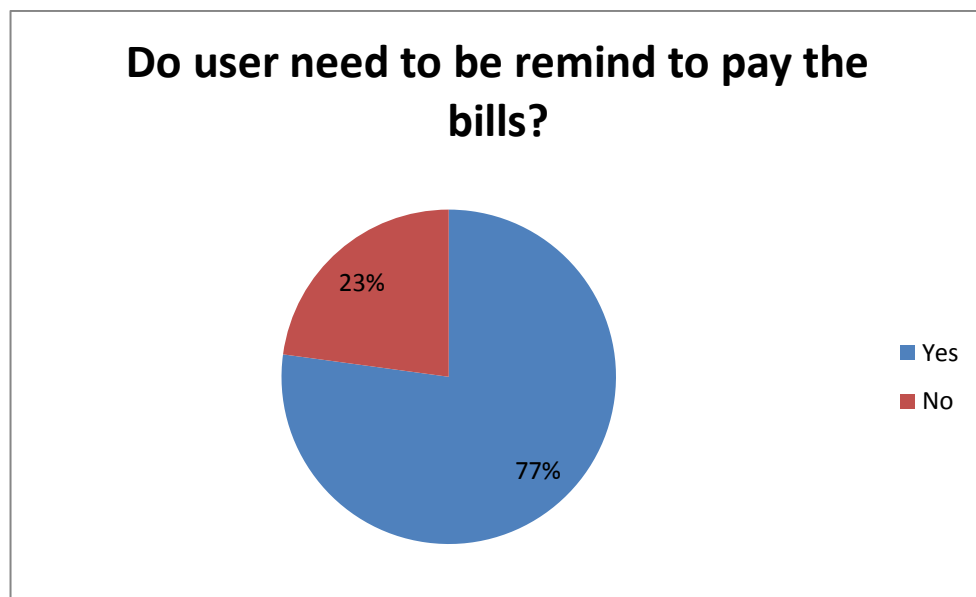


FIGURE 9. Result finding do really user need to be remind to pay the bills

Lastly, based from the results, the author found that 77% user need help in order to remind them about the bills need to be pay. Only 23% does not help to remind to pay the bills. The author strongly belief that SAPPNA will able to solve all problems that the user faced and ease their daily life process.

3.1.3 Existing Mobile Applications

The author also had collected the information from the internet to gain knowledge about the existing mobile application in the current market. Three existing mobile application were choose which is Bill Reminder, Easy Bills Reminder and Mobile Recharge and Bill Pay. The description of the exiting mobile application and comment was properly recorded for reference.

3.2 System Development

In order to develop this project, there are few method suggested to develop a mobile application. The author decided to use prototype methodology model in this project. The prototype model is one of model for software development lifecycle model in which software prototype created for demonstration and requirement elaboration. In evolutionary prototype model, it includes four main phases which are definition basic requirements, creating the working prototype, verification of the working prototype and changing or elaboration the requirements. **The idea of choosing prototype model because as the project still on going, it ensures more changes can be made in order to meet the user requirement for the final product.**

With this model, it help to ensure the functionality based on the requirements that consumer are looking from this mobile application. As soon as the project is near the completion, the prototypes should already be well calibrated with the requirements and needs of its target market. This method segment has been segmented into five main phase. The steps involved are shown as below;

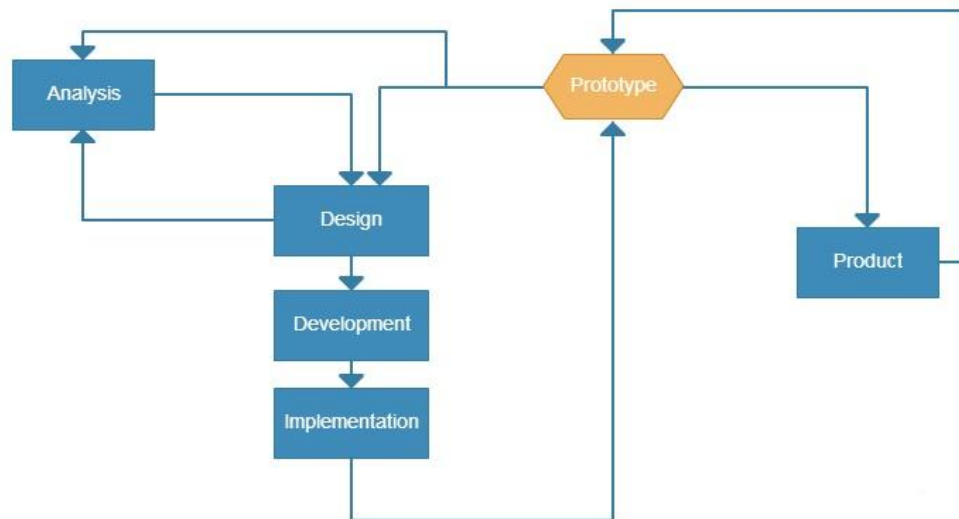


FIGURE 10. Prototype Model

3.2.1 Analysis Phase

During the analysis phase, all important information related to the background of the project has been collected and gathered as described in Chapter 1. In the first stage of the project, the author defined the problem that lead to the idea of developing an application. All the requirements are gathered through observations, research and questionnaire regards to the problem.

The scope of the project has been carefully decided during the requirement phase. The author also comes out with feasibility study which includes technical, economy and operational feasibility. The first stage of the project was very important because it is the main core before the author develops the application. Other than that, author also develops a Gantt chart to display the entire task that will be carried out through the project lifecycle (refer Appendices).

All the information gathered in previous phase was analysed in this phase. All the published papers, earlier research and journals on related topic, title and scope of project were studied by the author. The author also considers and examines any techniques that have been used by the other developer to be considered in order developing the application later. From the finding, there are a lot of existing applications that are quite similar to this project. All those application acted as a guide and benchmark for the author in developing the application. In order to develop the application, the author also makes some studies and comparison between the available platforms that can be suit to develop a mobile application.

3.2.2 Design and Development Phase

At this phase, all the diagrams, interfaces, flow and storyboard of the project were described in details. The design and function of each screen of the application need to be properly structured so that the user will able to have good

experience while using the application later. The application should not be too complicated and must be user-friendly.

Coding part also include in this phase to develop the application. The phase involves the programming and coding function and features of the application. To develop this application, author used android studio for software and SQLite database. For hardware used were laptops and android tablet.

3.2.3 Implementation Phase

In implementation phase, the author started to develop the application. The prototype of the application was built based on the analysis and design set in earlier previous phase. If there are any changes that found, mistakes or recommendations, the changes will be done to the prototype at this phase. The author needs to ensure the prototype will be functioning and can be sued during the usability testing as well as all the function in the application should be work.

During this phase also, the author conduct unit testing for few features in the application. The purpose of unit testing is to determine whether it run exactly what that has been already planned.

TABLE 2. Results from the Unit Testing

Features	Testing	Result	Remarks
Login	Correct Username & Password	Successful	
	Incorrect Username & Password	Unsuccessful	
	Correct Username & Incorrect Password	Unsuccessful	
	Incorrect Username & Correct Password	Unsuccessful	

Logout	Click the Logout button	Need to key in the Username and Password	Logout button allow the user to exit the application securely
Profile	Enter the data during register	Data enter will be the same during register	
Register	Incomplete details	Unsuccessful	All the details needed need to be complete

For SAPPNA the author only tests the certain important features of the application, Login, Logout, Profile and Register.

For Login features, if the user key in the correct username and password, the user able to login into the application. If the user key in incorrect user name and password or correct username and incorrect password else vice versa, the user unable to proceed to the next page.

Next, Logout features. For this features it does allow the user to exit the application securely because if the user want to get back to the application, the user need to key in again the username and password.

For Profile, the data that has been enter during the register if the user creates a new account it will be show the same in the profile page as the user want to review the details entered.

Lastly, if the user enter incomplete details during register, the user unable to proceed to the next stage. The user needs to key in the details and complete the registration process.

3.2.4 Prototype Phase

At the last stage of this project, System Usability Scale test were carried out to the user in order to identify either the application meet all the requirements needed. Feedbacks and comments from the user were analysed and will be take into consideration in order to improving the quality of the application.

For working prototype, it will take a lot of time to develop it. The task will be repeated until the system prototype function well and ensure the coding can run efficiently and effectively.

3.3 System Architecture

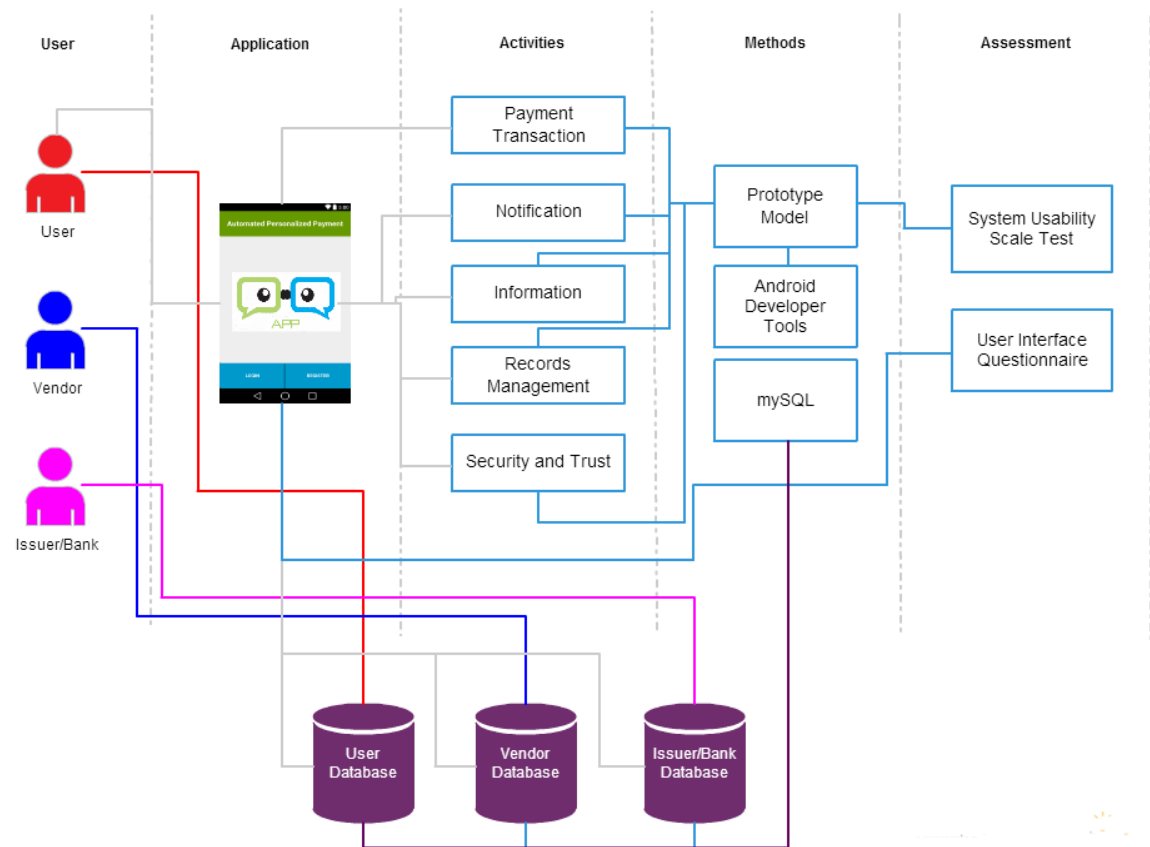


FIGURE 11. System Architecture for SAPPNA

From the proposed system architecture shown in Figure 11, it describes the whole design of the SAPPNA. The system starts with the user. The user is the main actor in this application because user is the one using the application, the vendor and the bank does not interact in using the application. The system starts from the user where user will download and install the application into their smartphone.

There are five activities that interact directly to the user. The activities are the Payment Transaction, Notification, Information, Records Management, and Security and Trust.

In the 'Method' section, to develop the application, author need to understand what the methodology need to be use. The methodology used is Prototype Model. Besides, the platform will be used to develop the application are the Android Developer Tools and MySQL for the database.

After the development part complete, the prototype application will be test and assess by suing the System Usability Scale Test to make sure the user requirement can be fulfil and satisfied.

Lastly, this system need to have databases to stores all the information. For this prototype, author use virtual database to represent bank and vendor. For the user, author creates another database to store the information.

3.4 System Flow

System flow is way of displaying data flows in a system and how the decisions are made to control the events. The system flow was explain in the graphical representation of the flow of data in the system and represents the work process in the application system that later will be develop.

3.4.1 Use Case Diagram

A use case diagram will describe the interaction among the elements of a system. It is a methodology used in system analysis in order to identify, clarify

and organize the user requirements. The actor act as individuals who will involve in using the application and the use case are the specific roles played by the actor within the application.

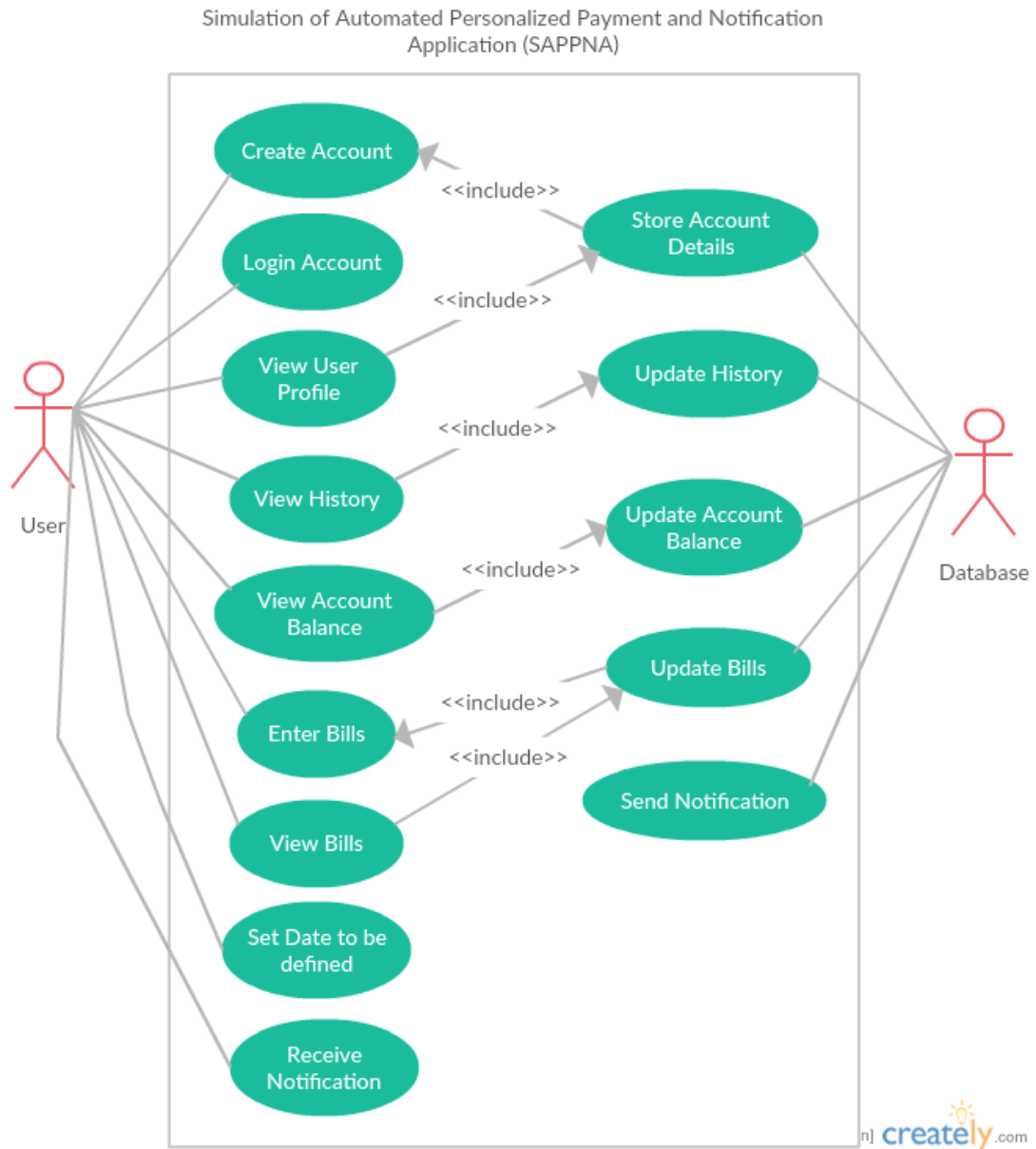


FIGURE 12. Use Case Diagram of SAPPNA

Referring to the use case diagram in Figure 12, the user and SAPPNA is the actor in this use case. The actor who is the user interacts with all the use case in the application. In this prototype, only the user will interact and get involve with the process and no include the bank or the issuer which is the vendor.

3.4.2 Entity Relationship Diagram

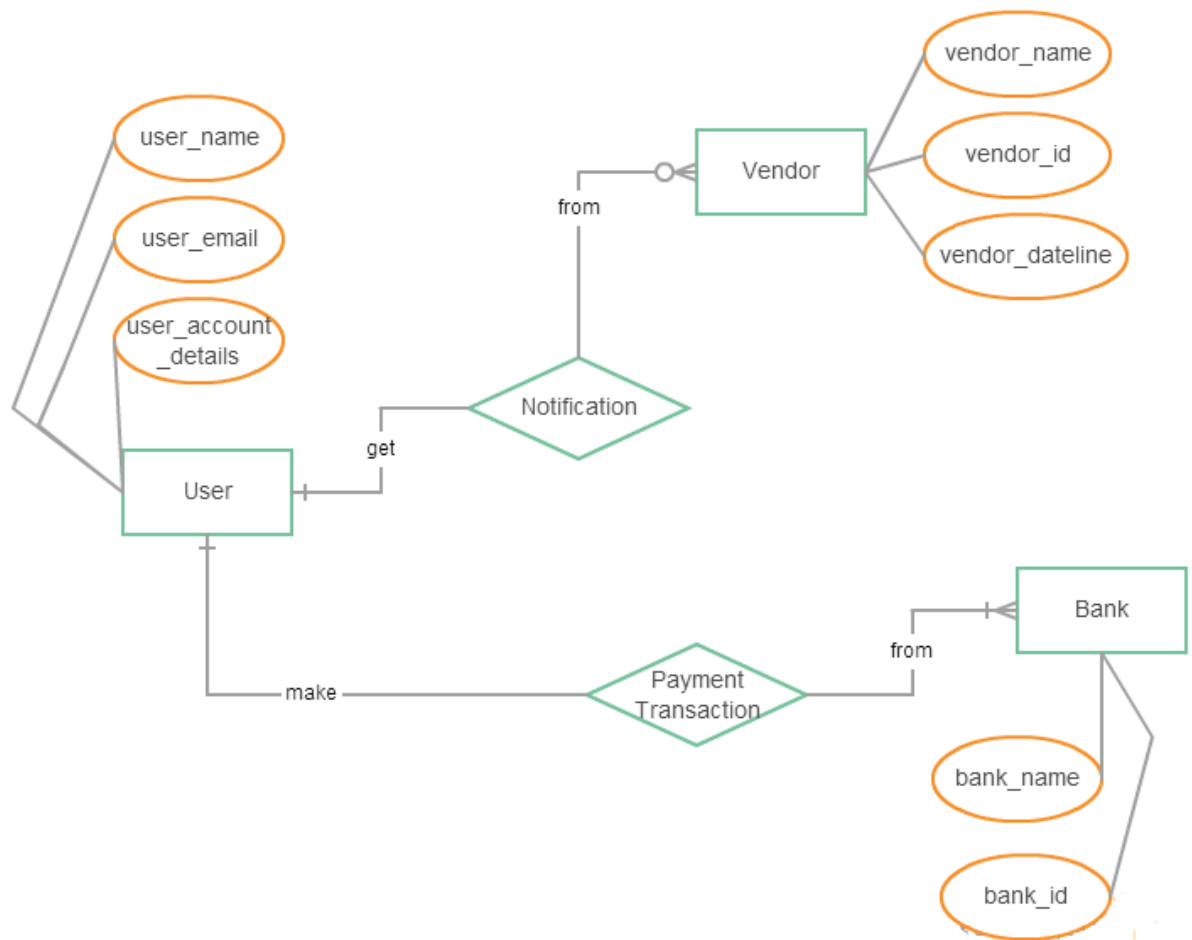


FIGURE 13. Entity Relationship Diagram of SAPPNA

The entity relationship diagram is a data modelling technique that describes in graphically illustrates the entities and the relationship between it. It is crucial to develop the entity relationship diagram because it helps in creating the database for the system application. It is using the high level logical data model, and will be useful in developing conceptual design for database.

In SAPPNA, the author will create databases for user, bank and the vendor. All the database need required data to make sure the application can run smoothly. For user database, the data needed in developing the initial prototype are the user name, user email and the user account details. Next, the database for bank,

the bank name and the bank id need. Lastly, the vendor's database, the vendor name, vendor id and vendor dateline. The relationship between the user and the bank is the bank will make the payment transaction for the user. For user and the vendor, the relationship is the vendor will give the notification to the user and the dateline and the bills need to be pay.

3.4.3 Data Flow Diagram

In data flow diagram, is a diagrammatic representation of the information movements within the system. Data flow diagram display how the information in and out of the system, what changes the information and where the information are saved.

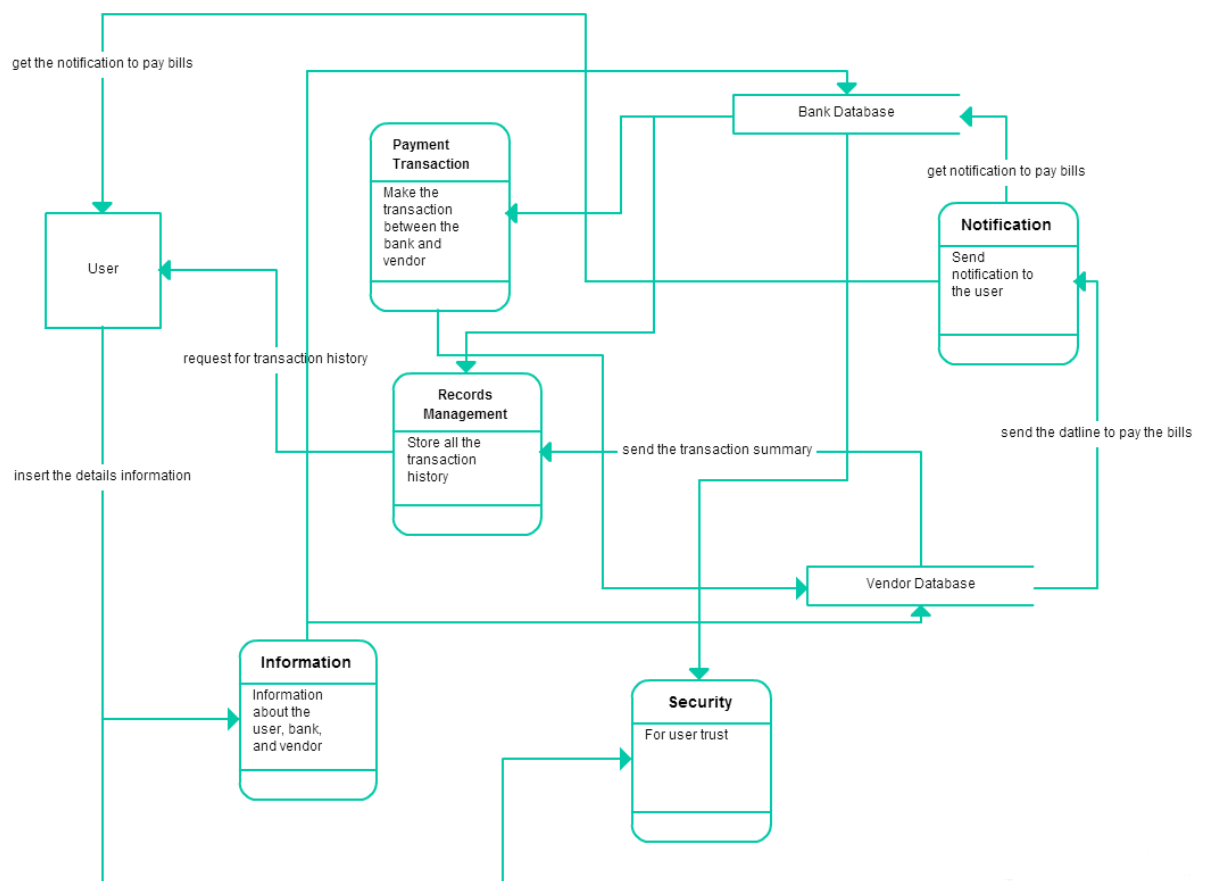


FIGURE 14. Data Flow Diagram of Application

From the Figure 14 the user, bank and vendor act as the database and also the source. The user sends data out by inserting the information. The user will

receive the output; then notification to pay bills and request for transaction history. For the vendor, the database receives the input as well sending the output. The vendor will send the deadline to pay the bills and send the transaction summary. For input part, it will receive the information of the user.

Next the bank, also received the input and sending the output. For input, the bank also gets the information from the user. In the output part, the bank will complete the transaction to the vendor.

CHAPTER 4

RESULTS AND DISCUSSION

In this chapter, the author will explain about the results and discussion from system design, interview results and the interface design. Usability testing was conducted to get the feedback from the prototype. In the interface design part, the author explained briefly how the application looks like.

4.1 System Usability Scale Test

Usability testing is a technique that been used to evaluate the application or system by the user. In usability testing, the testers will test the ease with which the user interface can be used. The test comprises the testing either the application is user friendly or not. Besides, usability test help to disclose either the users feel happy with the application or the system according to the parameters.

Set of questionnaire were gave to the respondents that use and experience using the prototype. Basically usability testing refers to evaluating a product or service by testing it with representative user. The goal of usability test is to identify any usability problem, collect quantitative data and to determine the users' satisfaction with the product.

20 numbers of respondents were recorded answering the usability test questionnaires. Scale 1 is strongly disagree, Scale 2 is disagree, Scale 3 neither agree nor disagree, Scale 4 is agree and Scale 5 is strongly agree.

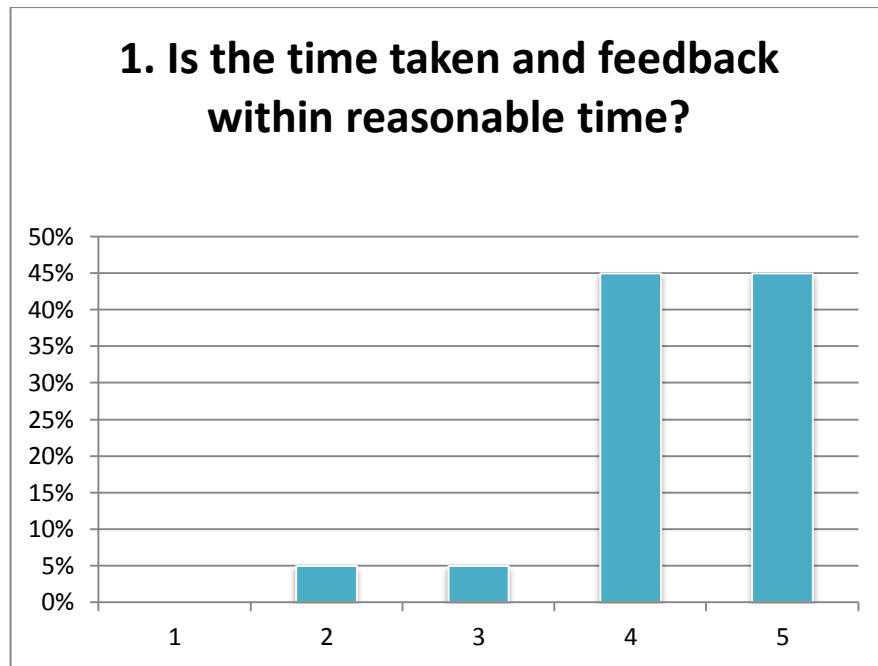


FIGURE 15. Results for the visibility of system status

The user shows that 45% strongly agree and 45% also agree that the time taken and feedback of the application is within the reasonable time. Only 5% agree neither agree nor disagree and the remaining 5% indicate disagree.

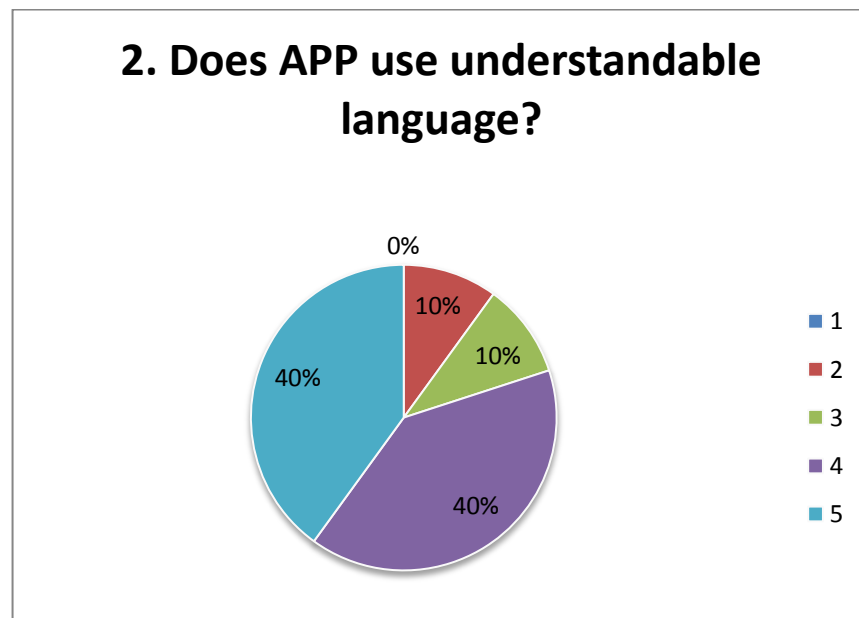


FIGURE 16. Results for match between system and the real world

From the above figure show the result for match between system and real world, thus the system speak the user language with words and concept that re familiar to the user? 40 % strongly agree and also the same amount of 40% agrees that SAPPNA use understandable language and only 10% neither agree nor disagree and 10 % disagree with the language used.

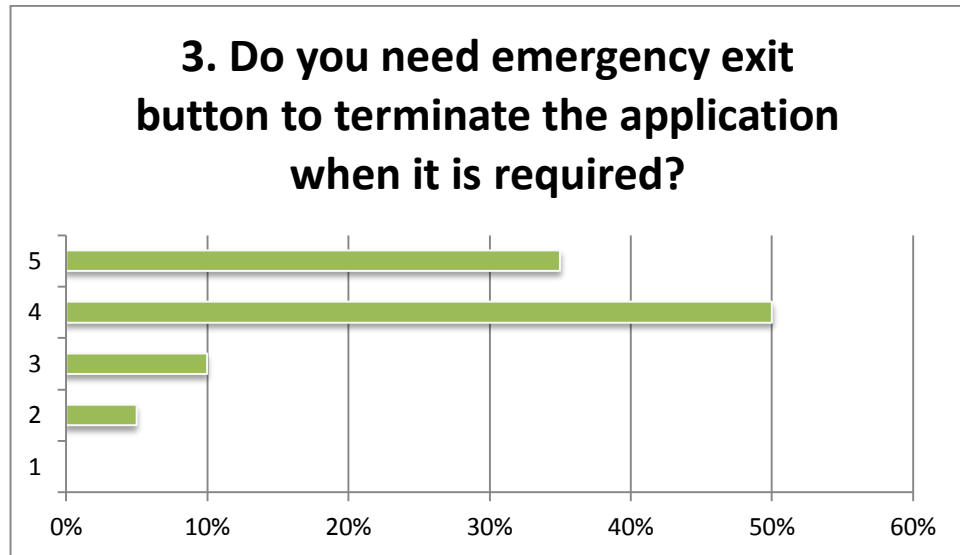


FIGURE 17. Results for user control and freedom

40% strongly agree that SAPPNA should have emergency exit button to terminate the application when it is required. For agrees, 40% of them under scale 4 and 20% neither agree nor disagree about the emergency exit button.

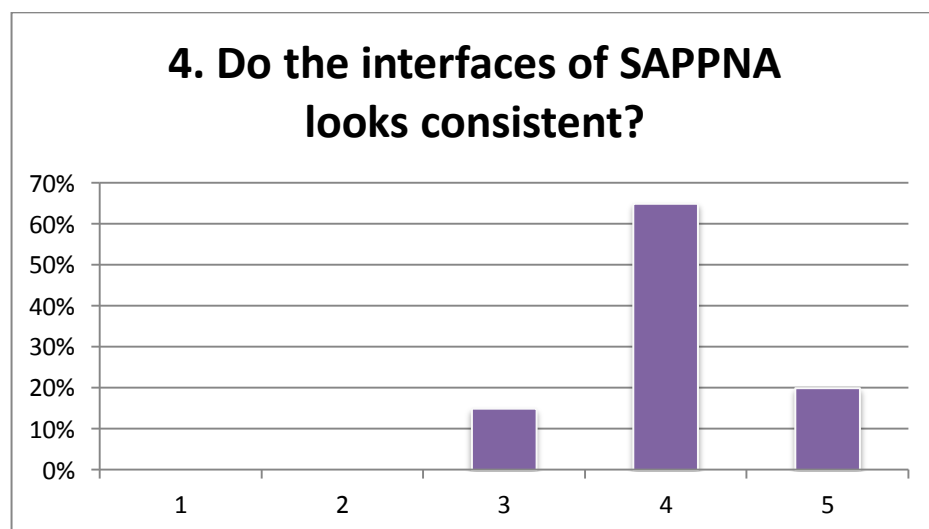


FIGURE 18. Results for consistency and standards

50% respondents agree that the interface of SAPPNA looks consistent. And only 20% strongly agree and 20% neither agree nor disagree about the interfaces.

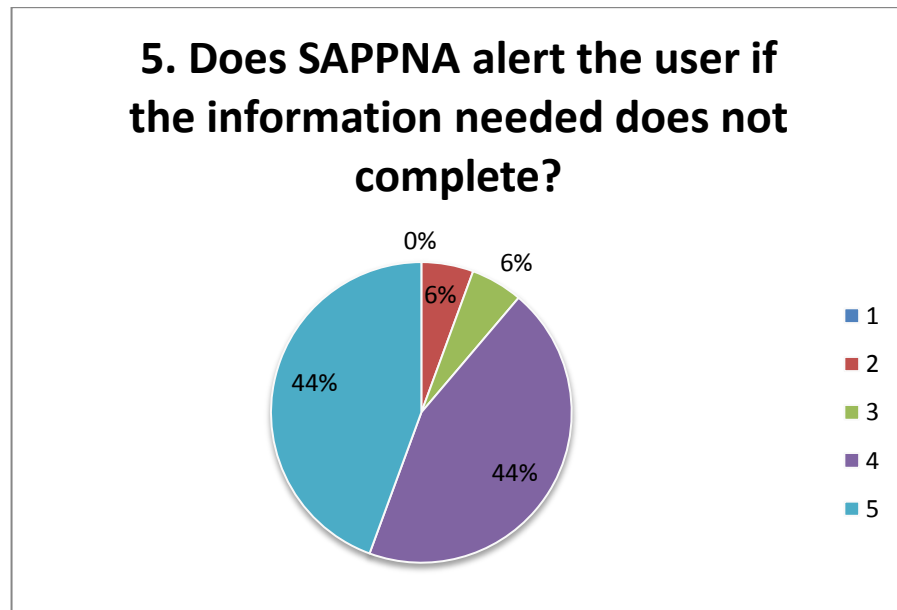


FIGURE 19. Results for error prevention

Error prevention help the user form making those errors when user the application. For SAPPNA, 78% out of 22% agree that SAPPNA need the alert for the user if the information needed to use the application is incomplete.

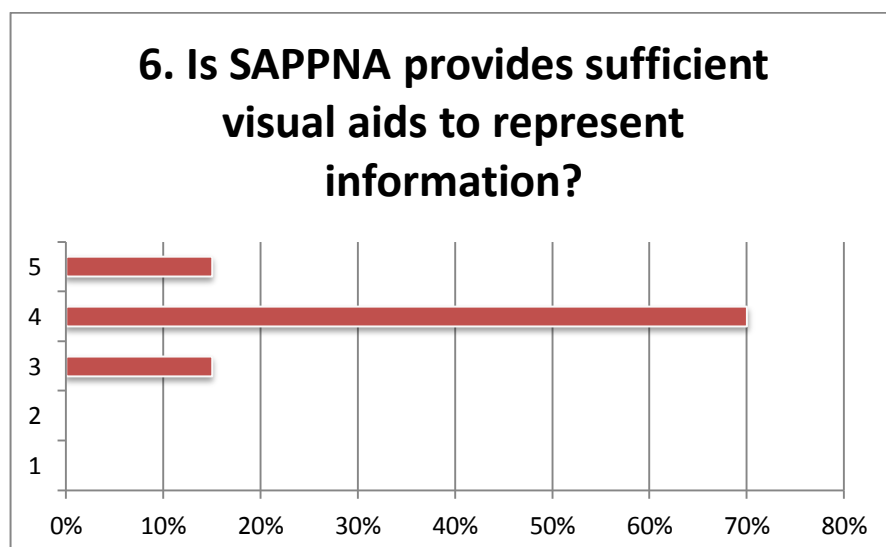


FIGURE 20. Results for recognition or recall

Recognition or recall helps to minimize the user's memory load by making object, actions and options more visible. For SAPPNA 20% respondents

strongly agree that SAPPNA provides sufficient visual aids to represent information. While 70% agree and 10% neither agree nor disagree.

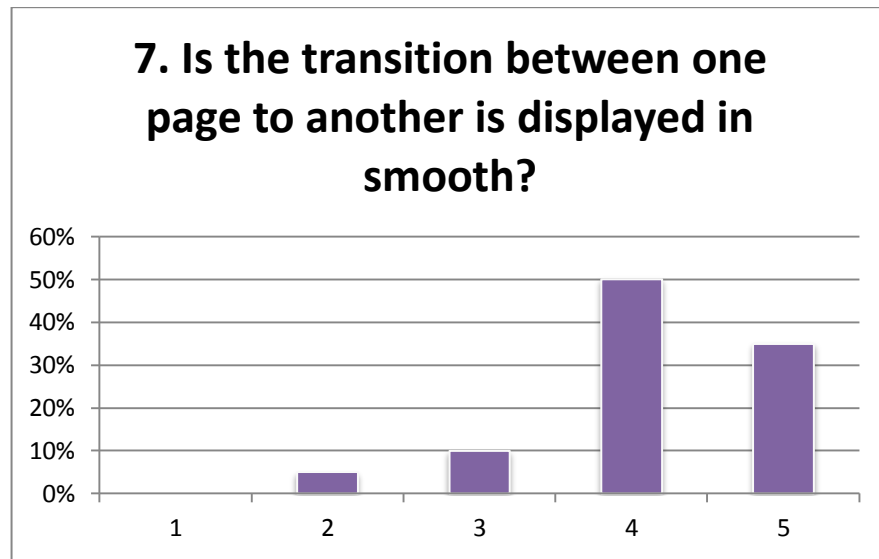


FIGURE 21. Results for flexibility and efficiency of use

From Figure 21, the testing result it shows that 60% respondents agree that the transition of the page to another is displayed in smooth while 20 % strongly disagree and 20% neither agree nor disagree.

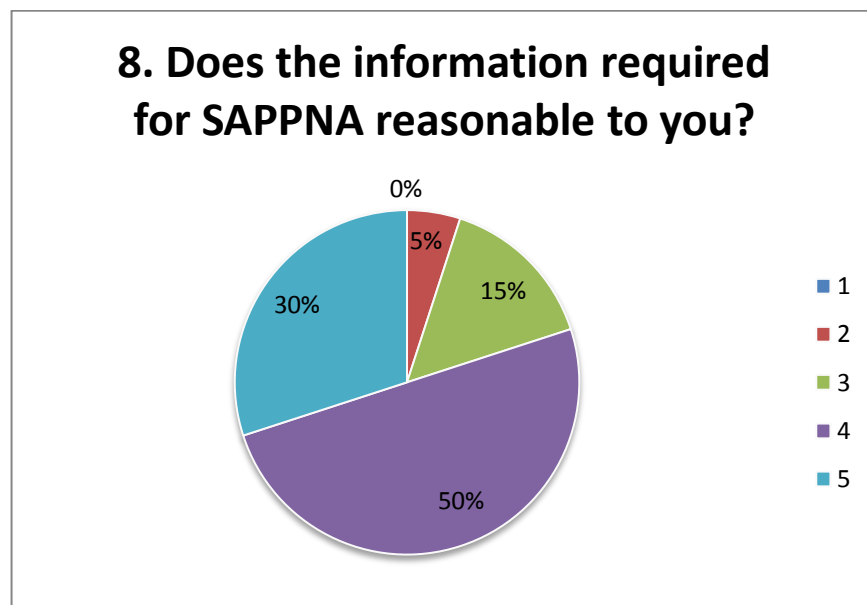


FIGURE 22. Results for aesthetic and minimalist design

50% respondents agree the information for SAPPNA that needs to be enter by the user are reasonable while 40% strongly agree. For neither agree nor disagree only 10%.

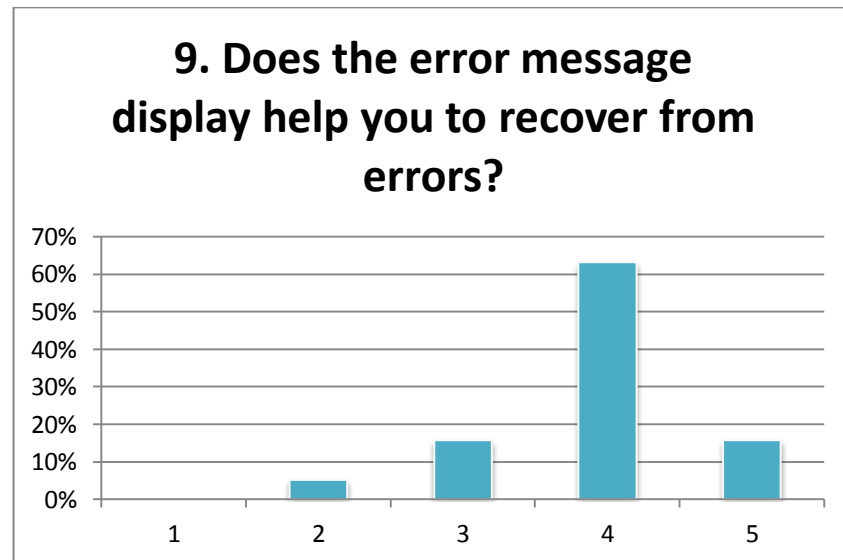


FIGURE 23. Results for help users recognize diagnose and recover from errors.

For error message, it should be expressed in plain language and precisely indicate the problem. 10% strongly agree and 60% agree the error message that had been display help them to recover form errors. The remaining, 10 % disagree while 20% neither agree nor disagree.

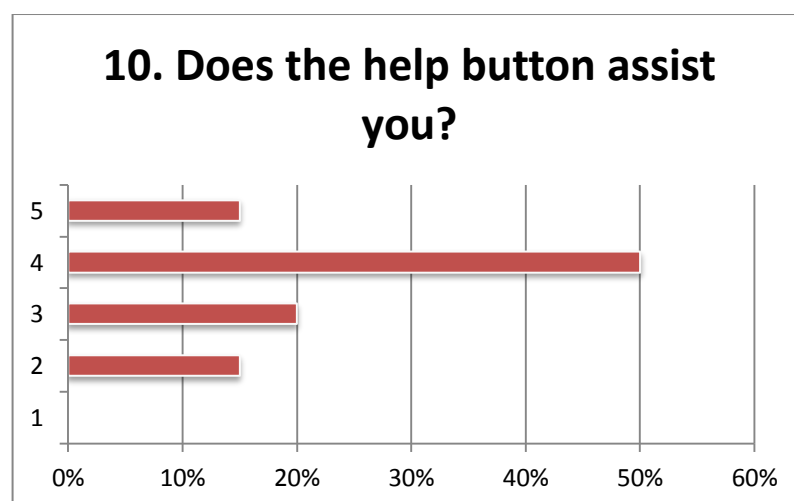


FIGURE 24. Results for help and documentation

Help button will assist the user regards to the application as 10% of the respondents strongly agree. 60% of the respondents agree that the help button assist them when using SAPPNA. 10% neither agree nor disagree and the balance 20% disagree that the help button can assist them.

4.2 Interface Design

Below are the interface designs of SAPPNA:

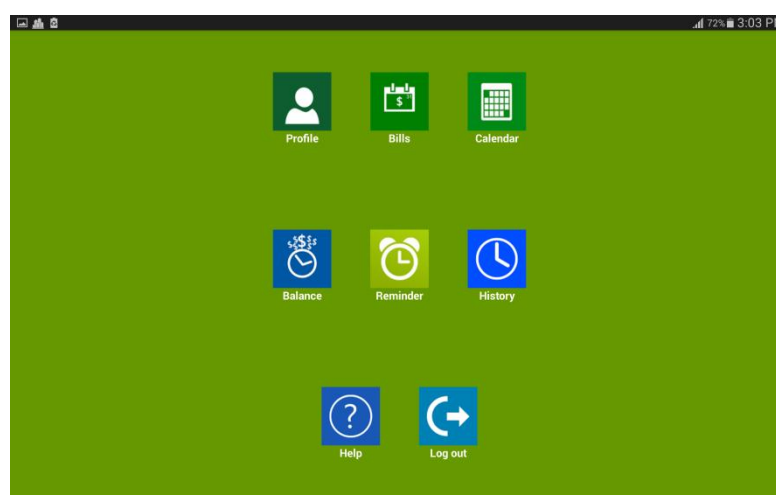


FIGURE 25. The User Home

The user home include the Profile, Bills, Calendar, Bank, Reminder, History, Help and Logout icon

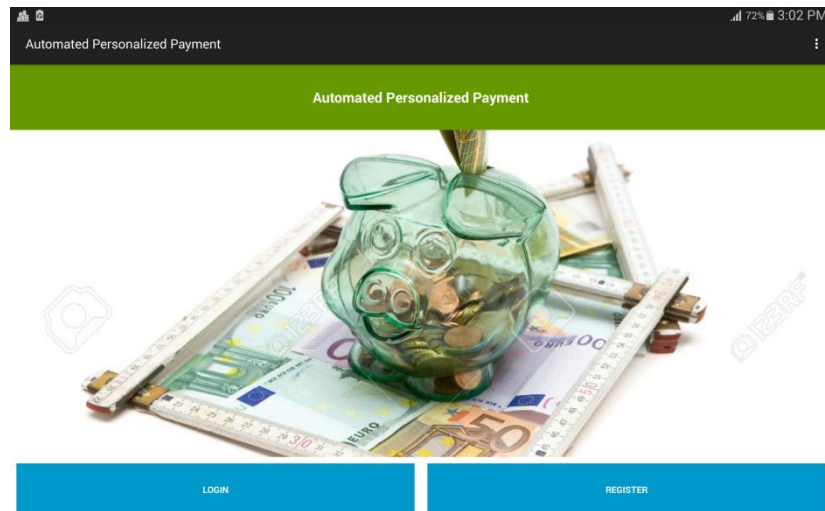


FIGURE 26. The Home Page

Two activities that user need to choose either login or register.

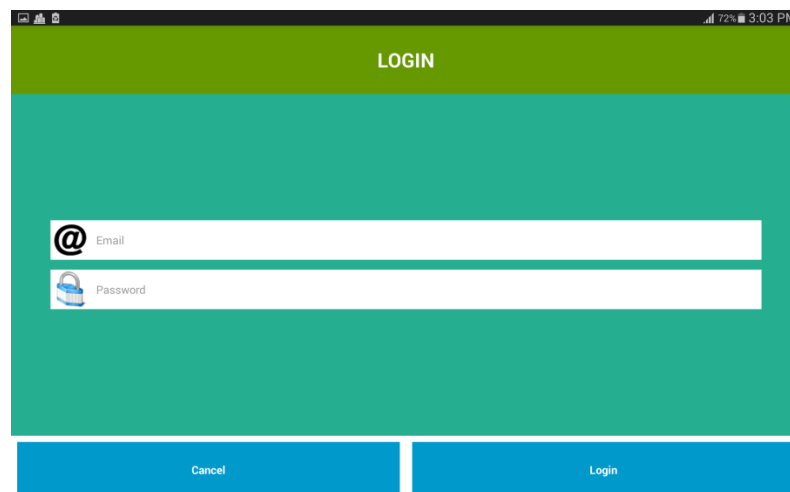
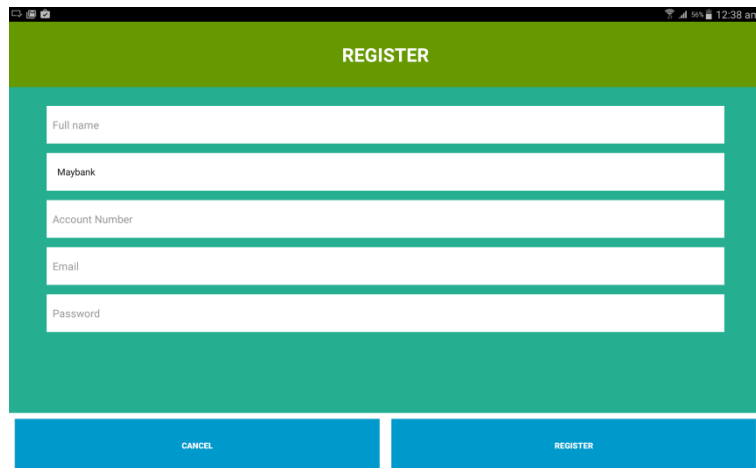


FIGURE 27. The Login Page

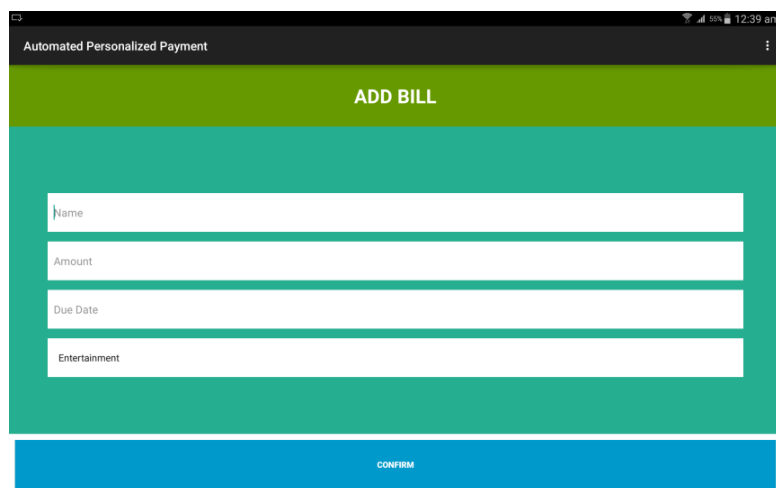
If the users choose the login in the Figure 27, the user will enter the email as user name and the password. The user then will choose to login using the account or cancel the process.



The screenshot shows a mobile application interface for a registration page. At the top, there is a green header bar with the word "REGISTER" in white capital letters. Below the header, there is a teal background area containing five white input fields stacked vertically. The labels for these fields are "Full name", "Maybank", "Account Number", "Email", and "Password". At the bottom of the teal area, there are two blue buttons: "CANCEL" on the left and "REGISTER" on the right.

FIGURE 28. The Register Page

If the users choose register, the user needs to register the account to use the application. The user needs to enter the Name, Email, Password and Confirmed Password. If the user want to proceed, user can choose Register and if not Cancel.



The screenshot shows a mobile application interface for an "Add Bill" page. At the top, there is a green header bar with the text "Automated Personalized Payment" in white. Below this, there is a teal background area with a green header bar containing the text "ADD BILL" in white. Below the header, there are four white input fields stacked vertically with labels "Name", "Amount", "Due Date", and "Entertainment". At the bottom of the teal area, there is a single blue button labeled "CONFIRM".

FIGURE 29. The Add Bill Page

Page where user can add the bills that need to be pay. The user needs to enter the Name of Bill, the Amount, the Due Date and choose the categories of the bill.

CHAPTER 5

CONCLUSION

The application's objective to study how a mobile payment can be used to ease the user life and implement and develop an android mobile application that can be used by the user for Simulation of Automated Personalized Payment and Notification Application (SAPPNA). The scopes of the system application are the automatic transaction payment and notification to the user.

SAPPNA will be developing in prototype first before the real product is put to the public. The prototype will have many versions in order to fulfil the user requirement. After all the stage of prototype model complete, the prototype will be tested out and identify any new requirement. If there any addition in the requirement of the application system, new version of prototype will be develop. Once all the requirements have been fulfil, the real product will be produce.

Besides, in future enhancement the application will have system administrator who will supervise and handle the security and the database. The system administrator will be another actor who interacts with the application besides the user. The application will have payment gateway that will help user in actual transaction payment with the bank.

Lastly, the project will be able to solve the problem that the user faces. As the application will be able to give notification to the user when the bills dateline as well as the user will able to have automatic transaction payment for their bills.

REFERENCES

- Bell, K. (2014, August 3). Android Overtakes iOS in Usage Stats for the First Time Ever. Retrieved from <http://www.technobuffalo.com/2014/08/03/android-overtakes-ios-in-usage-stats-for-the-first-time-ever/>
- Dahlerg, T., Mallat, N., Ondrus, J., & Zmijewska, A. (2007, September 9). Past, present and future of mobile payments research. *Electronic Commerce Research and Applications*.
- Holzer, A. & Ondrus, J.(2010, May 18). Mobile application market: A developer's perspective
- Ion, F. (2014, September 23). 10 Android features that still make it better than iOS 8. Retrieved from <http://www.greenbot.com/article/2686006/10-android-features-that-still-make-it-still-better-than-ios-8.html>
- Jahns, R. (2010, May 17). 6 reasons why mobile apps will become as important for companies as corporate websites. Retrieved from <http://research2guidance.com/corporate-app-will-become-as-important-for-companies-as-having-a-corporate-website-a-look-at-the-german-market/>
- McCracken H. (2013, April 16). Who's winning, iOS or Android? All the numbers, all in one place. Retrieved on July 24, 2013 from <http://techland.time.com/2013/04/16/ios-vs-android/>
- Pirzadeh, K., & Kekicheff, M. B. (2010, September 28). Mobile Payment Application Architecture. *United States Patent Application Publication*.
- Pousttchi, K. (2003). Conditions for acceptance and usage of mobile payment procedures. Munich Personal RePEc Archive.

Pousttchi, K. (2007, July 21). A modeling approach and reference models for the analysis of mobile payment use cases. *Electronic Commerce Research and Applications*.

Rehman, S. u., & Coughlan, J. (2013). An Efficient Mobile Payment System Based On NFC Technology. *World Academy of Science, Engineering and Technology*.

APPENDICES

Gantt Chart and Key Project Milestone

Final Year Project 1: January 2015

ID	Task	W1	W2	W3	W4	W5	W6	W7	W8	W9	W10	W11	W12	W13	W14
1	Final Year Project 1														
2	Selection of Project Title														
3	Preliminary Research Work														
4	Problem Identifying and Literature Review														
5	Submission of Extended Proposal														
6	Project Research and Pre Development														
7	Project Analysis and Synthesis														
8	Proposal Defence														
9	Project Work Continues														
10	Submission of Interim Report														

Final Year Project 2: May 2015

ID	Task	W1	W2	W3	W4	W5	W6	W7	W8	W9	W10	W11	W12	W13	W14
1	Final Year Project 2														
2	Project Development														
3	Submission of Progress Report														
4	Project Development continues														
5	Pre-SEDEX														
6	Submission of Draft Report														
7	Submission of Dissertation (Soft Bound)														
8	Submission of Technical Paper														
9	Viva														
10	Submission of Dissertation (Hard Bound)														

W	Week
	FYP Progress
	Task Progress
	Milestone/Dateline

Database Class

```
public class SQLiteHandler extends SQLiteOpenHelper {

    private static final String TAG = SQLiteHandler.class.getSimpleName();

    // All Static variables
    // Database Version
    private static final int DATABASE_VERSION = 1;

    // Database Name
    private static final String DATABASE_NAME = "nadia_db";

    // Login table name
    private static final String TABLE_USER = "user";

    // Login Table Columns names
    private static final String KEY_ID = "id";
    private static final String KEY_FULL_NAME = "name";
    private static final String KEY_BANK_ACCOUNT_NAME =
"bank_account_name";
    private static final String KEY_BALANCE = "balance";
    private static final String KEY_EMAIL = "email";
    private static final String KEY_PASSWORD = "password";

    public SQLiteHandler(Context context) {
        super(context, DATABASE_NAME, null, DATABASE_VERSION);
    }

    // Creating Tables
    @Override
    public void onCreate(SQLiteDatabase db) {
        String CREATE_USER_TABLE = "CREATE TABLE IF NOT EXISTS " +
TABLE_USER + " ( "
            + KEY_ID + " INTEGER PRIMARY KEY AUTOINCREMENT, " +
KEY_FULL_NAME + " TEXT, " + KEY_BANK_ACCOUNT_NAME
            + " TEXT, " + KEY_BALANCE + " TEXT, " + KEY_EMAIL + " TEXT, "
            + KEY_PASSWORD + " TEXT)";
        db.execSQL(CREATE_USER_TABLE);
        Log.d(TAG, "Database tables created");
    }

    // Upgrading database
    @Override
    public void onUpgrade(SQLiteDatabase db, int oldVersion, int newVersion)
    {
        // Drop older table if existed
        db.execSQL("DROP TABLE IF EXISTS " + TABLE_USER);
        // Create tables again
        onCreate(db);
    }

    //check if email is already existed
    public boolean isEmailExisted(String email){
        String countQuery = "SELECT * FROM " + TABLE_USER + " WHERE email
= '" + email + "'";
        SQLiteDatabase db = this.getReadableDatabase();
        Cursor cursor = db.rawQuery(countQuery, null);
        int rowCount = cursor.getCount();
        if(rowCount == 1){
            db.close();
            cursor.close();
        }
    }
}
```

```

        return true;
    }
    db.close();
    cursor.close();
    // return row count
    return false;
}

//add user if email is not found
public void addUser(String name, String bankAccountName, String
balance, String email, String password) {
    SQLiteDatabase db = this.getWritableDatabase();
    ContentValues values = new ContentValues();
    values.put(KEY_FULL_NAME, name);
    values.put(KEY_BANK_ACCOUNT_NAME, bankAccountName);
    values.put(KEY_BALANCE, balance);
    values.put(KEY_EMAIL, email); // Email
    values.put(KEY_PASSWORD, password);
    // Inserting Row
    long id = db.insert(TABLE_USER, null, values);
    db.close(); // Closing database connection
    Log.d(TAG, "New user inserted into sqlite: " + id);
}

//isLogin
public boolean loginUser(String email, String password){
    String countQuery = "SELECT * FROM " + TABLE_USER + " WHERE email
= '" + email + "' AND password = '" + password + "'";
    SQLiteDatabase db = this.getReadableDatabase();
    Cursor cursor = db.rawQuery(countQuery, null);
    int rowCount = cursor.getCount();
    if(rowCount == 1){
        db.close();
        cursor.close();
        return true;
    }
    db.close();
    cursor.close();
    // return row count
    return false;
}

public User getUserByEmail(String email){
    User user = new User();
    String selectQuery = "SELECT * FROM " + TABLE_USER + " WHERE
email='" + email + "'";
    SQLiteDatabase db = this.getReadableDatabase();
    Cursor cursor = db.rawQuery(selectQuery, null);

    //loop through all rows
    if(cursor.moveToFirst()){
        do {
            user.setName(cursor.getString(1));
            user.setBankAccountName(cursor.getString(2));
            user.setBalance(cursor.getString(3));
            user.setEmail(cursor.getString(4));
            user.setPassword(cursor.getString(5));
        }while (cursor.moveToNext());
    }
    cursor.close();
    db.close();
    return user;
}

//add cash by email
public int addCashByEmail(String email, String amount){
    SQLiteDatabase db = this.getWritableDatabase();
    ContentValues values = new ContentValues();

```



```

        values.put(KEY_BALANCE, amount);
        return db.update(TABLE_USER, values, KEY_EMAIL + " = ?", new
String[]{email});
    }
}

```

User Class

```

public class User {

    private String name;
    private String bankAccountName;
    private String balance;
    private String email;
    private String password;

    public User(){}

    public User(String name, String bankAccountName, String balance, String
email, String password){
        this.name = name;
        this.bankAccountName = bankAccountName;
        this.balance = balance;
        this.email = email;
        this.password = password;
    }

    public String getName() {
        return name;
    }

    public void setName(String name) {
        this.name = name;
    }

    public String getBankAccountName() {
        return bankAccountName;
    }

    public void setBankAccountName(String bankAccountName) {
        this.bankAccountName = bankAccountName;
    }

    public String getBalance() {
        return balance;
    }

    public void setBalance(String balance) {
        this.balance = balance;
    }

    public String getEmail() {
        return email;
    }

    public void setEmail(String email) {
        this.email = email;
    }
}

```

```

    public String getPassword() {
        return password;
    }

    public void setPassword(String password) {
        this.password = password;
    }
}

```

Notification function

```

switchAlloc = savedKey.getPref("switch", getApplicationContext());
date = (DatePicker) findViewById(R.id.datePicker);
submit = (Button) findViewById(R.id.submit);
toggle = (Switch) findViewById(R.id.switchNotification);

NotificationManager notificationManager =

(NotificationManager) getSystemService(Context.NOTIFICATION_SERVICE);
notificationManager.cancelAll();

if (switchAlloc.equals("on")) {
    toggle.setChecked(true);
    toggle.setText("Notification ON");
} else {
    toggle.setChecked(false);
    toggle.setText("Notification OFF");
}

toggle.setOnCheckedChangeListener(new
CompoundButton.OnCheckedChangeListener() {
    public void onCheckedChanged(CompoundButton buttonView, boolean
isChecked) {
        if (isChecked) {
            toggle.setText("Notification ON");
            savedKey.putPref("switch", "on", getApplicationContext());
            final Intent intent = new Intent();
            intent.setClass(getApplicationContext(),
ActivityServices.class);
            //startService(intent);
        } else {
            toggle.setText("Notification OFF");
            savedKey.putPref("switch", "off", getApplicationContext());
            NotificationManager notificationManager =
(NotificationManager)
getSystemService(Context.NOTIFICATION_SERVICE);
            notificationManager.cancelAll();
        }
    }
});

submit.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        if (toggle.isChecked()) {

```

```

        Intent intent = new Intent();
        Calendar calendar = Calendar.getInstance();
        calendar.set(Calendar.MONTH, date.getMonth());
        calendar.set(Calendar.DAY_OF_MONTH, date.getDayOfMonth());
        calendar.set(Calendar.YEAR, date.getYear());
        calendar.set(Calendar.HOUR_OF_DAY, 8);
        calendar.set(Calendar.MINUTE, 30);

        AlarmManager alarmManager = (AlarmManager)
        getSystemService(Context.ALARM_SERVICE);

        intent.setClass(getApplicationContext(),
        ActivityServices.class);
        PendingIntent pendingIntent =
        PendingIntent.getService(getApplicationContext(),
        0, intent, PendingIntent.FLAG_UPDATE_CURRENT);
        alarmManager.set(AlarmManager.RTC_WAKEUP,
        calendar.getTimeInMillis(), pendingIntent);
    }
}
});

```